FREE eBook

LEARNING Windows

Free unaffiliated eBook created from
Stack Overflow contributors.

#windows

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: windows

It is an unofficial and free Windows ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Windows.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with Windows

Remarks

Windows is a series of operating systems developed and published by software company Microsoft.

you can get expanded information from Wikipedia - Microsoft Windows

Versions

Desktop versions

Version	Date
DOS	1981-08-01
1.01	1985-11-20
1.02	1986-05-01
1.03	1986-08-01
1.04	1986-04-01
2.03	1987-12-09
2.10	1988-05-27
2.11	1989-03-13
3.0	1990-05-22
3.1	1992-04-22
NT 3.1	1993-06-27
Workgroups 3.11	1993-11-01
3.2	1993-11-22
NT 3.5	1994-09-22
NT 3.51	1995-05-30
95	1995-07-24

Version	Date
NT 4.0	1996-07-24
98	1998-06-25
2000	2000-02-17
ME	2000-08-14
XP	2001-10-25
XP Professional x64	2005-04-25
Vista	2007-01-30
7	2009-10-22
8	2012-10-26
8.1	2013-10-17
10	2015-07-29
10 Anniversary update	2016-08-01

Server versions

Version	Date
NT 3.1	1993-08-01
NT 3.5	1994-09-01
NT 3.51	1996-06-14
NT 4.0	1995-07-29
2000	2000-02-17
Server 2003	2003-04-01
Server 2003 R2	2006-12-01
Server 2008 R2	2009-07-01
Server 2012	2012-08-01
Server 2012 R2	2013-10-01

Version	Date
Server 2016	2016-09-01

Mobile versions

Version	Date
Pocket PC 2000	2000-04-19
Pocket PC 2002	2001-10-01
Mobile 2003	2003-06-23
Mobile 2003 SE	2003-03-24
Mobile 5.0	2005-05-09
Mobile 6.0	2007-01-03
Mobile 6.1	2008-04-01
Mobile 6.5	2009-05-11
Phone 7	2010-10-21
Phone 7.5	2011-09-27
Phone 7.8	2013-02-01
Phone 8	2012-10-29
Phone 8.1	2014-08-15
10 Mobile	2016-11-20

Examples

Information

The History

Microsoft windows is an operating system available in 137 languages, wrote in C,C++ & Assembly , can be installed on "ARM, IA-32, Itanium, x86-64, DEC Alpha, MIPS, PowerPC" platforms

first release was in November 20 ,1985 as windows 1.0

with about 2 billion PC(personal computer) windows is the most used operating system

Read Getting started with Windows online: https://riptutorial.com/windows/topic/1710/getting-started-with-windows

Chapter 2: Programming Custom Desktop File/Folder context menu extensions

Examples

Using Registry entries

This works well for single level right click context menu. All you need to do is create a registry entry under Classes Root HKEY_CLASSES_ROOT for specific extension. If you want to create a entry for all types of files choose * else choose extension like .pdf etc.

```
var regmenu = Registry.ClassesRoot.CreateSubKey("*\\shell\\hello");
if (regmenu != null)
    regmenu.SetValue("", "Hello World");
var regcmd = Registry.ClassesRoot.CreateSubKey("*\\shell\\hello\\command");
    if (regcmd != null)
         regcmd.SetValue("", "Do something" );
```

This works well for one level menus

Simple example is at http://www.codeproject.com/KB/cs/appendmenu.aspx?msg=2236729

Using ShellSharp

When you need multi level menus, with multiple parameters SharpShell comes to rescue. https://github.com/dwmkerr/sharpshell has umpteen number of examples and it works perfect even for single level to multi level custom context menus.

Key thing is to create class with attributes [ComVisible(true)] and [COMServerAssociation(AssociationType.AllFiles)] and inheriting class fromSharpContextMenu which implements CanShowMenu and CreateMenu functions and you need to register the assembly via regasm tool or ServerRegistrationManager that Sharpshell creator recommends

```
[ComVisible(true)]
[COMServerAssociation(AssociationType.AllFiles)]
public class AdvancedContextMenu : SharpContextMenu
{

    protected override bool CanShowMenu()
    {
        //  We can show the item only for a single selection.
    }
    protected override ContextMenuStrip CreateMenu()
    {
        //  Create the menu strip.
        var menu = new ContextMenuStrip();
        ... add any level of ToolStripMenuItems and add them to menu
        return menu
    }
```

```
}
```

More details can be obtained at https://github.com/dwmkerr and http://www.codeproject.com/Articles/512956/NET-Shell-Extensions-Shell-Context-Menus

Read Programming Custom Desktop File/Folder context menu extensions online: https://riptutorial.com/windows/topic/5423/programming-custom-desktop-file-folder-context-menu-extensions

Chapter 3: Serial Ports

Introduction

Using the serial ports on Windows can be a bit complex. This documentation section will, in time, explain all about the use of DCBs, CreateFile(), port events, and asynchronous serial communication.

Examples

Listing all serial ports.

Getting all serial ports information from Windows is often necessary, you may want to give the user a choice of ports to open, or check if your device is connected.

In addition, some ports just cannot be opened using a "COMx" string and need to be opened using a device name. Some older versions of Windows cannot open ports named 'COMxx' when the port number is 10 or higher. So, using the device number is a sensible way to identify your com port with Windows.

Serial port information is held in the registry under the key
`HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\SERIALCOMM`

This example in C shows how to list all the serial ports and how one can get to the device name to use for the call to CreateFile.

The method is quite simple:

- Open the registry at `HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\SERIALCOMM`
- Enumerate all keys we find there. The device name is the key name, and the 'display name' is the key value.

As is often the case with Windows programming, most of the effort is spent on error checking. It may seem a bit silly, but there's no safe way to avoid it.

```
#define WIN32_LEAN_AND_MEAN  // excludes stuff frokm windows.h that we won't need here.
#include <Windows.h>
#include <string.h>
#include <tchar.h>
#include <malloc.h>

void ShowErrorFromLStatus(LSTATUS lResult)
{
    LPTSTR psz;
    FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        lResult,
        0,
        (LPTSTR)&psz,
```

```
        1024,
        NULL);

    _tprintf(_T("Windows reports error: (0x%08X): %s\n"), lResult, (psz) ? psz :
_T("(null)"));
    if (psz)
    {
        LocalFree(psz);
    }
}

int main()
{
    DWORD nValues, nMaxValueNameLen, nMaxValueLen;
    HKEY hKey = NULL;
    LPTSTR szDeviceName = NULL;
    LPTSTR szFriendlyName = NULL;
    DWORD dwType = 0;
    DWORD nValueNameLen = 0;
    DWORD nValueLen = 0;
    DWORD dwIndex = 0;

    LSTATUS lResult = RegOpenKeyEx(HKEY_LOCAL_MACHINE, L"HARDWARE\\DEVICEMAP\\SERIALCOMM", 0,
KEY_READ, &hKey);
    if (ERROR_SUCCESS != lResult)
    {
        printf("Failed to open key \'HARDWARE\\DEVICEMAP\\SERIALCOMM\' \n");
        ShowErrorFromLStatus(lResult);
        return 1;
    }

    lResult = RegQueryInfoKey(hKey, NULL, NULL, NULL, NULL, NULL, NULL,
        &nValues, &nMaxValueNameLen, &nMaxValueLen, NULL, NULL);

    if (ERROR_SUCCESS != lResult)
    {
        _tprintf(_T("Failed to RegQueryInfoKey()\n"));
        ShowErrorFromLStatus(lResult);
        RegCloseKey(hKey);
        return 2;
    }

    szDeviceName = (LPTSTR)malloc(nMaxValueNameLen + sizeof(TCHAR));
    if (!szDeviceName)
    {
        _tprintf(_T("malloc() fail\n"));
        RegCloseKey(hKey);
        return 3;
    }

    szFriendlyName = (LPTSTR)malloc(nMaxValueLen + sizeof(TCHAR));
    if (!szFriendlyName)
    {
        free(szDeviceName);
        _tprintf(_T("malloc() fail\n"));
        RegCloseKey(hKey);
        return 3;
    }

    _tprintf(_T("Found %d serial device(s) registered with PnP and active or available at the
moment.\n"), nValues);
```

```
    for (DWORD dwIndex = 0; dwIndex < nValues; ++dwIndex)
    {
        dwType = 0;
        nValueNameLen = nMaxValueNameLen + sizeof(TCHAR);
        nValueLen = nMaxValueLen + sizeof(TCHAR);

        lResult = RegEnumValueW(hKey, dwIndex,
            szDeviceName, &nValueNameLen,
            NULL, &dwType,
            (LPBYTE)szFriendlyName, &nValueLen);

        if (ERROR_SUCCESS != lResult || REG_SZ != dwType)
        {
            _tprintf(_T("SerialPortEnumerator::Init() : can't process registry value, index:
%d\n"), dwIndex);
            ShowErrorFromLStatus(lResult);
            continue;
        }
        _tprintf(_T("Found port \'%s\': Device name for CreateFile(): \'\\.%s\'\n"),
szFriendlyName, szDeviceName);
    }

    free(szDeviceName);
    free(szFriendlyName);
    RegCloseKey(hKey);
    return 0;
}
```

Program output on my laptop:

```
Found 1 serial device(s) registered with PnP and active or available at the moment.
Found port 'COM23': Device name for CreateFile(): '\.\Device\BthModem0'
```

Read Serial Ports online: https://riptutorial.com/windows/topic/10700/serial-ports

Credits

S. No	Chapters	Contributors
1	Getting started with Windows	Community, Derpcode, duskwuff, H. Pauwelyn, Mark Hurd, Pro-Fun, Pseudonym Patel
2	Programming Custom Desktop File/Folder context menu extensions	Pooran
3	Serial Ports	Michaël Roy